Amendments To The Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A processor-implemented method for exposing an application object model to web-based clients, comprising:

parsing each hyper-text transport protocol (HTTP)-form-processing request received from a client by a web server;

traversing objects in the object model in response to each property name**[[s]]** obtained from the HTTP-form-processing request; **[[and]]**

wherein traversing includes comparing a property name from the request to a property name of an object defined in the object model; and

returning to the client a value of a selected property of an object in the object model in response to traversal of the object model.

2. (Original) The method of claim 1, further comprising initiating execution of a method of an object in response to a method name obtained from an HTTP-form-processing request.

3. (Original) The method of claim 1, further comprising:

creating an event queue in response to a first keyword obtained from an HTTP-form-processing request;

registering, in response to a second keyword obtained from an HTTP-form-processing request, with the application objects to receive events generated by the application objects;

entering the events in the event queue; and

returning events in the event queue to the client in response to a third keyword obtained from an HTTP-form-processing request.

4. (Original) The method of claim 1, further comprising aborting parsing of an HTTP-form-processing request in response to an annotated property associated with a property name obtained from the HTTP-form-processing request.

5. (Original) The method of claim 1, further comprising converting a character string parameter in an HTTP-form-processing request to a value having a data type compatible with a property of a property name obtained from the HTTP-form-processing request and setting the property of the object to the value.

6. (Original) The method of claim 1, further comprising, for a reference-type property of an object found in traversing the object model, generating a unique identifier for the reference-type property, registering the unique identifier for the reference-type property in a collection, and returning a Uniform Resource Identifier (URI) that includes the unique identifier to the client.

7. (Original) The method of claim 6, wherein the URI is a globally unique identifier (GUID).

8. (Original) The method of claim 6, further comprising maintaining each identifier in the collection according to a lifetime type associated with each identifier, wherein a short-lifetime type reference is controlled by client access to the short-lifetime type reference, and a long-lifetime type reference is controlled by an application object.

9. (Original) The method of claim 8, further comprising removing a short-lifetime type reference from the collection if a client does not submit an HTTP-forms-processing request containing the URI for a selected period of time.

10. (Original) The method of claim 9, further comprising maintaining long-lifetime type reference in the collection until a referenced application object is removed.

11. (Original) The method of claim 1, further comprising:

configuring the web server arrangement with an initial object name associated with the a name of an application object model; and

retrieving an object definition of the initial object name in response to the name of the application object model obtained from an HTTP-form-processing request.

12. (Currently Amended) A computing arrangement for exposing objects to web-based clients, comprising:

a web server arrangement that includes a data processing system configured with a web server;

an object model library hosted by the web server, the object model library containing the objects to be exposed;

a web objects model hosted by the web server, the web objects model configured to parse each hyper-text transport protocol (HTTP)-form-processing request received from a client by a web server, traverse objects in the object model in response to property names obtained from the HTTP-form-processing request, wherein traversing includes comparing a property name from the request to a property name of an object defined in the object model, and return to the client a selected property of an object in the object model in response to traversal of the object model.

13. (Original) The arrangement of claim 12, wherein the web objects model is further configured to initiate execution of a method of an object in response to a method name obtained from an HTTP-form-processing request.

14. (Original) The arrangement of claim 12, wherein the web objects model is further configured to initiate a method to create an event queue in response to a first keyword obtained from an HTTP-form-processing request, and initiate a method to register, in response to a second keyword obtained from an HTTP-form-processing request, with the application objects to receive events generated by the application objects, and return events in the event queue to the client in response to a third keyword obtained from an HTTP-form-processing request.

15. (Original) The arrangement of claim 12, wherein the web objects model is further configured to abort parsing of an HTTP-form-processing request in response to an annotated property associated with a property name obtained from the HTTP-form-processing request.

16. (Original) The arrangement of claim 12, wherein the web objects model is further configured to convert a character string parameter in an HTTP-form-processing request to a value having a data type compatible with a property of a property name obtained from the HTTP-form-processing request and set the property of the object to the value.

17. (Original) The arrangement of claim 12, wherein the web objects model is further configured to generate, for a reference-type property of an object found in traversing the object model, a unique identifier for the reference-type property, registering the unique identifier for the reference-type property in a collection, and return a Uniform Resource Identifier (URI) that includes the unique identifier to the client.

18. (Original) The arrangement of claim 17, wherein the URI is a globally unique identifier (GUID).

19. (Original) The arrangement of claim 17, wherein the web objects model is further configured to maintain each identifier in the collection according to a lifetime type associated with each identifier, wherein a short-lifetime type reference is controlled by client access to the short-lifetime type reference, and a long-lifetime type reference is controlled by an application object.

20. (Original) The arrangement of claim 17, wherein the web objects model is further configured to remove a short-lifetime type reference from the collection if a client does not submit an HTTP-forms-processing request containing the URI for a selected period of time.

21. (Original) The arrangement of claim 20, wherein the web objects model is further configured to maintain long-lifetime type reference in the collection until a referenced application object is removed.

22. (Original) The arrangement of claim 12, further comprising:

an initial object name associated with the a name of an application object model in the web server; and

wherein the web objects model is further configured to retrieve an object definition of the initial object name in response to the name of the application object model obtained from an HTTP-form-processing request.

23. (Currently Amended) An apparatus for exposing an application object model to web-based clients, comprising:

means for parsing each hyper-text transport protocol (HTTP)-form-processing request received from a client by a web server;

means for traversing objects in the object model in response to property names obtained from the HTTP-form-processing request and for comparing a property name from the request to a property name of an object defined in the object model as part of the traversing; and

means for returning to the client a selected property of an object in the object model in response to traversal of the object model.

24. (Original) The apparatus of claim 23, further comprising:

means for creating an event queue in response to a first keyword obtained from an HTTP-form-processing request;

means for registering, in response to a second keyword obtained from an HTTP-form-processing request, with the application objects to receive events generated by the application objects; and

means for returning events in the event queue to the client in response to a third keyword obtained from an HTTP-form-processing request.